Balance trucks: Using crowd-sourced data to procedurally-generate gameplay within mobile games

Mark Lewis, Sylvester Arnab, Luca Morini, Samantha Clarke, Lorenz Klopfenstein, Alessandro Bogliolo, Saverio Delpriori and Alex Masters

Original citation & hyperlink:
Lewis, Mark Russell, et al. "Balance Trucks: Using Crowd-Sourced Data to Procedurally-Generate Gameplay within Mobile Games." *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*. IEEE, 2018.

https://dx.doi.org/10.1109/VS-Games.2018.8493412

DOI 10.1109/VS-Games.2018.8493412

Publisher: IEEE

Balance Trucks: Using Crowd-Sourced Data to Procedurally-Generate Gameplay within Mobile Games

Mark Russell Lewis, Sylvester Arnab, Luca Morini, Samantha Clarke, Alex Masters
Disruptive Media Learning Lab. (DMLL)
Coventry University
Coventry, United Kingdom

Lorenz Klopfenstein, Alessandro Bogliolo, Saverio Delpriori
Department of Pure and Applied Sciences (DiSPeA)
University of Urbino
Urbino, Italy

***Abstract*—Within the field of procedural content generation (PCG) research, until now, the use of crowd-sensing data has primarily been used as a means for the collection of information and the generation of feedback relating to player experience within games, and game aesthetics [1], [2]. However, crowd-sensing data can offer so much more, supplying a relatively untapped font of information, and inspiration, which might be used within the creation of unique PCG game spaces or content, whilst providing a highly visible outlet for the dissemination of crowd-sensed material to users. This paper examines one such use of crowd-sensed data, the creation of a gamification layer for the CROWD4ROADS (C4RS) [3] application, SmartRoadSense (SRS) [4]. The authors will open with a brief discussion of PCG. Following this, an explanation of the features and aims of the SmartRoadSense application will be provided. Finally, the paper will introduce 'Balance Trucks', the SmartRoadSense gamification layer, discussing the concepts behind using crowd-sensed data within its design, its development and use of PCG.**

Keywords—gamification; procedural content generation; game design; video game; racing game, crowd-sensing

I. INTRODUCTION

The art of making videogames is a multidisciplinary process that is extremely expensive in both cost and effort. A high-quality triple-A title can require thousands of person hours to develop, and production budgets often stretch to millions of pounds. This expense is gradually increasing over time, as new generations of technology; a new console or graphics card, are released, making the requirements for developing a title ever more complex. It therefore seems logical that developers have a strong need to generate tools and systems which will help to reduce the production costs and effort put into the creation of videogame content [5].

Procedural content generation (PCG) within videogames can be described as the use of a computers processing power to create content via mathematical equations using small amounts of data, or data which has been obtained indirectly from users, as inputs. Togelius, Kastbjerg, Schedl and Yannakakis simplify this definition, describing PCG as "the algorithmical creation of game content with limited or indirect user input" [6]. The use of PCG provides several interesting benefits for the development of commercial games. The creation of tools which can generate specific content for use within a game, after some modification by a designer or artist, has become relatively widespread within the industry; particularly when used with artificial intelligence or the generation of environments. PCG's ability to provide infinite gameplay supplies an interesting method for increasing replayability. This is seen as an important factor within any videogame as it is believed that if a title has a high replay value, players will remain engaged and continue to play it [7]. These two uses of PCG, the use of tools and the ability to offer endless gameplay, reduce the effort required to construct a videogame and, therefore, help to lower production costs [8]. PCG also provides a highly visible platform for the content which is used to create the game [9]. This is important to the authors of this paper, who believe that this statement could be extended to include the data which is used to create the content on display.

The first uses of PCG within videogames came about due to a need to compress data and resolve issues surrounding the lack of storage capacity and processing power found within early home computers [7]. "Rogue" (1980) [10] is often quoted as being the first computer game to utilise PCG as a means for generating playable content. Indeed, it went on to spawn a whole genre of RPG titles which are now referred to as roguelike games. Arguably the most famous early exemplar of PCG is Elite (1984) [11], which used a single seed number to generate eight complete galaxies, each containing 256 solar systems including planets, space stations, and economies that the player could explore and interact with. Since the 1980s', PCG has been used for niche roles, sporadically, in games [8]. However, the last several years have seen a resurgence in its use within commercial games. This may be due to what could be described as 'the Minecraft effect'. In 2011, Minecraft [12], which utilises a seed number from the system clock to procedurally generate a seemingly infinite gameplay environment, saw massive commercial success and reinvigorated how developers look at, and use, PCG within their games. Recently, titles such as No Man's Sky [13] and the forthcoming Star Citizen [14] have moved the use of PCG

up another gear, generating galaxies, solar systems, and planets in a similar manner to the original Elite, but with a scale and detail that has never been seen before.

Within academia, the early use of PCG within videogames was mostly ignored, and interactions between the developers who were utilizing it and academics were almost nonexistent. In the last 15-20 years, academic interest in the subject has increased dramatically [8]. Researchers have since addressed a range of PCG topics. Several authors have attempted to define unified taxonomies for PCG [15], [16], and develop metaphors which aid the understanding of design relationships with the use of PCG [17]. Research has also been conducted into the practical uses of content generation. These studies have examined diverse uses for PCG, including subjects such as the generation of levels for platform games [18], [19], the creation of missions for action adventure games [20], the generation of entire worlds [21] and even attempts to build software that can procedurally generate complete games, from the ground up [22]. Recently, focus of PCG research and development has been directed at examining how complete levels, that are able to adapt to the players inputs, and thus playing style, might be created [23], [24].

In this paper, we describe the use of data gathered via crowd-sensing technology to procedurally generate gameplay levels and content for inclusion within a gamified layer of the SmartRoadSense (SRS) application [4]. To our best knowledge, this represents the first time that crowd-sensing data has been utilised as a seed for generating gameplay and disseminating that information back to the user.

The gamification of SmartRoadSense is aimed at fulfilling four specific criteria: 1) the dissemination of crowd-sensing data back to users of the application, 2) an increase in the applications user-base, 3) the promotion of car-sharing and car-sharing initiatives, 4) the provision of a platform which can be utilised for gamification purposes by other projects funded by the EU Horizon 2020 [25] research and innovation programme.

In the following sections, we will first provide an overview of SmartRoadSense, describing how the application gathers and utilises road surface data. We will then discuss the design and development of our gamification layer, titled 'Balance Trucks', and examine how it uses crowd-sensing data collected by its parent application, SmartRoadSense, to create gameplay and content in a manner which disseminates the data they have gathered back to them. Finally, we will examine the challenges that lie ahead for Balance Trucks as it progresses towards completion and offer up our conclusions about the project.

II. SmartRoadSense

Developed by an international team, led primarily by the University of Urbino, Italy, as part of CROWD4ROADS (C4RS) [3], an EU Horizon 2020 funded project, SmartRoadSense is a crowd-sensing system for the continuous monitoring of road quality. The application promotes active citizenship and allows volunteers to freely participate in the collection of valuable road infrastructure data.

The system consists of three constituent components: i) an Android/iOS application, which collects data from the triaxial accelerometers and GPS sensors of an enabled smartphone and generates location-based estimates relating to the quality of a road's surface; ii) a cloud-based service onto which the data collected by the application is downloaded and aggregated with existing cartographic data from OpenStreetMap [26], and iii) a web-based user interface, which provides access to the data (distributed under an open license, in CSV format), and an interactive map showing a visual representation [4], [27].

The collection of SmartRoadSense data (see Fig.1) begins with the associated application. Once the app has been started by the user, it will quietly begin gathering data from the smartphone's accelerometers and GPS sensor as the user drives. No further input is required until the end of a journey, at which point the user can simply tap their device's screen to halt data collection. The gathered data is then processed within the smartphone to generate a single number for each second of the journey; a "roughness index" value, which estimates the quality of the road's surface at given points along the recorded track; the speed of the vehicle at each point; and a timestamp. Each data recording session is then given a unique identifier, a track ID, and automatically uploaded at the next available opportunity. Every 6 hours, new tracks are aggregated with the pre-existing data on the SmartRoadSense servers and mapped using OpenStreetMap data. Once this process is complete, a new set of roughness index data for each of the sampled locations is published for visualization and download on the SmartRoadSense website [4], [28], [27].

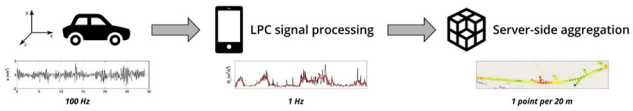

Fig.1. The process of collecting SmartRoadSense data.

Other applications have attempted similar aims as SmartRoadSense in the past, for example "Pothole Patrol" and "Nericell". In 2008, Pothole Patrol experimented with vehicles fitted with tri-axis accelerometers, GPS, and computers, to detect potholes; the experiment proved that of the anomalies detected, 90% needed repair [29]. A contemporary of Pothole Patrol, Nericell, utilised an array of smartphone sensors, including accelerometers, microphones, GSM radio, and GPS, to monitor traffic flow and road conditions [30]. Further examples of vehicular crowdsensing include: "VTrack" [31], which estimates possible traffic delays; "CTrack" [33], which aims to generating accurate location data based on GSM radio signals; "OpenSense" [33], a system which uses mobile sensors to monitor urban air pollution; "Wolverine" [34], which estimates traffic and road conditions; and several other studies making use of common smartphones to collect data related to road infrastructure [35], [36].

Whilst SmartRoadSense shares a lineage with each of these crowdsensing applications, its continuous monitoring system has been deployed for more than 3 years, collecting data thanks to the contributions of more than 3000 users. The system is being progressively scaled up, now being available to the public in several countries, including Italy, the United Kingdom, Romania, and Greece. It has already amassed a large

volume of data, with information gathered from over 18 million data points. This provides analysis for almost 52,000 kilometres of road surface; at time of writing. Within Italy, Regione Abruzzo is in discussions with local transport companies and municipalities with regards to the systematic adoption of SmartRoadSense, whilst Regione Marche intends to make use of the collected data when performing statistical analysis and data validation tests. In Romania, agreements have been made with local authorities who are interested in utilizing the application's data, whilst in the UK, Buckinghamshire Advantage have recently begun using the application on public transport and service vehicles.

III. Balance Trucks

The gamification layer of SmartRoadSense has been in development since January 2016. During this time, the concepts used within the gamification design have undergone periods of radical change. In its original format, the design consisted of a 'create your own story' adventure game [37]. This allowed players to create personalized characters before presenting them with a section of story. At the end of each section, the player would be required to select an action to perform in response to the storyline. The chosen action would determine an outcome for this section and define the next. This process continued until the storyline was complete. The player could then choose to replay a storyline which they had previously attempted or select a brand-new storyline instead.

Whilst this approach was proven to be perfectly valid during an early prototyping phase, it was remarked that the gamification did not help to disseminate information about the data which users would be collecting within SmartRoadSense back to them. The concept of using the data alongside procedural content generation methods to build gameplay levels was discussed and examined. This led the team to define three new gamification proposals; a scrolling shooter, a platform game and a racing game. Each of these were designed so that they could utilise PCG methods to generate associated playable environments, and gameplay. With the three proposals complete, they were disseminated to SmartRoadSense partners for feedback. It was then decided, by the entire team, that the gamification should be based on the racing game proposal as this was the most relevant option, providing a direct analogy for the data collection process, and the easiest proposal to develop. However, questions were later raised as to whether the gamifications multiplayer mode should take the format of a balancing game. To answer this, a small scale-survey which compared the two highest rated racing games on the Apple App Store against the two highest rated balancing games was conducted within Coventry Universities Disruptive Media Learning Lab (DMLL). This showed, that of the 24 students surveyed, 74% preferred racing games to balancing; making racing the obvious choice for multiplayer.

Therefore, in single-player mode, Balance Trucks is a 2D side-view racing game which sees the player being provided with a vehicle and then challenged to complete a form of time trial; transporting an object across each level in the fastest time possible. Based on the feedback received from case studies, Multiplayer mode does away with the use of objects and allows players to simply race against each other across the procedurally generated terrain. Throughout each level, players can collect coins, boosters and rare collectable car parts. Each of these are either procedurally generated, or procedurally placed, using randomization and PCG grammars, to provide additional subtlety and nuance to gameplay. In single-player, points are awarded to each player based upon their time to complete the level, whereas in multiplayer points are awarded based upon the players finishing position. The amount of points scored will be used to increase or decrease the player characters rank, this will in turn dictate the difficulty of the levels, as they are constructed, in single-player mode.

Within the gamification, the largest and arguably most important component to be procedurally generated is the playable terrain; the surface over which the players race. The procedural generation of Balance Trucks terrain involves the use of two forms of PCG: content selection, which is defined as the selection of content from a library, in this case SmartRoadSense data, and constraint satisfaction, which uses algorithms that apply constraints to the generated content [38]. Terrain is generated using 200 concurrent data entries, taken from the data collected by users within SmartRoadSense, and converting them into a height-map, which resembled a graph. The height-map is then modified to correct its scale and balance out any major spikes or troughs seen within the data. Further balancing is performed by comparing each of the used data entries against the sum of its neighbors divided by two. If the original numerical data is higher than the result of this sum, it will be replaced with the sum itself, otherwise the original figure will remain. If Data2 is the data being compared whilst Data1 and Data3 are its neighbors, this can be formulated as follows:

$$If\ Data2 > (Data1 + Data3) / 2\ then\ Data2 = (Data1 + Data3) / 2$$

$$If\ Data2 =< (Data1 + Data3) / 2\ then\ Data2 = Data2$$

Finally, to provide smooth terrain at the start and finish of each level, 10 new blocks of artificial data are added to the beginning and end of the SmartRoadSense data, making 20 fake blocks in total. The data which is added to the start of a level is generated using a random number between 0.0 and 0.3. The data between this and the first piece of genuine SmartRoadSense data is then generated using a similar equation to that used for balancing. Data which is added to the end of a level is generated in a similar manner, but in reverse; with the very last piece of level data being a randomly generated number and new data calculated between this and the final piece of SmartRoadSense data. If Data2 is the number being calculated, whilst Data1 and Data3 are its neighbors, the formulas would be as follows:

$$First\ and\ Last\ Data = Random\ Between\ 0.0\ and\ 0.3$$

$$Other\ Data = Data2 = Data1 + Data3$$

Once a playable surface has been generated, additional elements, coins, car components and boosters are added. Coins allow player to purchase items, such as boosters or additional vehicles, within the games offline store. Collecting special vehicle components unlock special vehicles, whereas boosters, which can be collected in any game mode but are only available for use within multiplayer, behave in a similar manner to the power ups found within Super Mario Kart [40]. Each of the collectibles will be procedurally-generated and placed into levels using PCG grammars, production rules, and algorithms, which generate content whilst dictating their frequency and availability. To provide an example of this, there is a 1:750 chance of a special car component being spawned onto one of the data points used to construct terrain; if a component is spawned this chance is reduced to zero for the remainder of the level.

With the terrain and additional elements generated, the complete level is skinned using a series of art assets. These have been created so that the level is gradually built up in a series of layers; a background, two mid-ground layers (one of which consists of the playable terrain) and a foreground. Each of the layers are placed slightly apart and behind the preceding layer to create a faux 3D parallaxing effect. This artwork has been generated by students from Coventry University's Faculty of Arts and Humanities under the banner of Phoenix Interactive; a Disruptive Media Learning Lab (DMLL) project, which teaches students about the games industry by involving them within the process of making games. A diagram showing the complete process of generating levels within Balance Trucks, the gamifications relationship with SmartRoadSense and its users is shown within Fig.2.

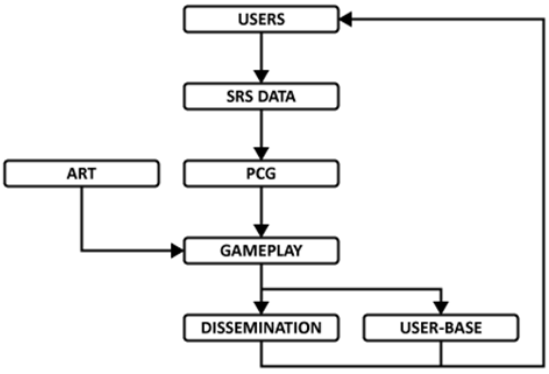

Fig.2. The interaction between users, data, procedural content generation, and gameplay.

At the end of each level, in both single and multiplayer, players will be presented with a range of information relating to their in-game performance. In single-player, this will include the time that the player takes to complete the level, the amount of points they score, the number of coins they collect, etc. In multiplayer, the information supplied will include the player's finishing position, the amount of points they have scored, and the amount of coins they have been awarded. Any points which the player gains will be added to the levelling system and the player will rank up accordingly, should their total pass the threshold for a level increase.

Within academic circles, there has been some discussion as to whether content selection offers sufficient complexity to warrant being called PCG [40]. The authors, however, agree with Smith, who states that "content selection, however simple, is a form of PCG when it is used to procedurally create an environment for the player to explore…" [39]. When used in combination with constraint selection, this provides the fastest method for generating content within Balance Trucks, and it allows SmartRoadSense data to be heavily reflected within the gameplay environment. This fulfils one of the EU 2020 criteria; that of disseminating SmartRoadSense data back to users.

IV. Conclusion

The SmartRoadSense gamification has undergone a long period of iteration and change. The design team did not set out with the aim of creating something which utilised PCG. Rather, the intention was to make the most fun and engaging game possible. However, whilst reviewing the gamification design, the team investigated how PCG could be used to create gameplay from SmartRoadSense data, which would also allow for the dissemination of data, which users had collected, back to them. This led to a complete re-examination, and thereafter re-design, of the gamification layer which would tie it more closely to its parent application.

By early November 2017, a new detailed design had been completed. This design utilises SmartRoadSense data, which has been gathered by the player themselves, and simple PCG methods as a basis for fun and engaging gameplay. Over the past few months, work on the project has moved into the development stage. The design and development of assets to reflect the levels and terrain associated to the SRS data has been a key focus. As part of the co-creative process to ensure variety in the art work that would provide assets for the PCG, students from the Faculty of Arts and Humanities were involved in the creation of the artwork for inclusion within the game. Several students were recruited and have since worked, alongside and tutored by the design team, to generate a range of artwork; including the gamification user-interface, vehicles and elements for use in the construction of procedurally-generated gameplay levels.

This process began with the creation of mock-ups showing how the gameplay environment and vehicles might look (Fig.3), plus mocks showing how each user-interface screen would work. Each of these is gradually being turned into full artwork for inclusion within the gamification (Fig.4). The on-screen display, sometimes called the HUD, has been finalised, barring any last-minute changes, and delivered to the code team in Urbino, alongside a range of other assets (Fig.5). Tests have also been run to determine how levels will be constructed and this has led to the realization that the utilization of a full testbed would be of great benefit to the team as development advances. Creation of this testbed is now in progress.

Fig.3. Mock-up of a Balance Trucks level, showing individual layers used to generate a parallax effect.

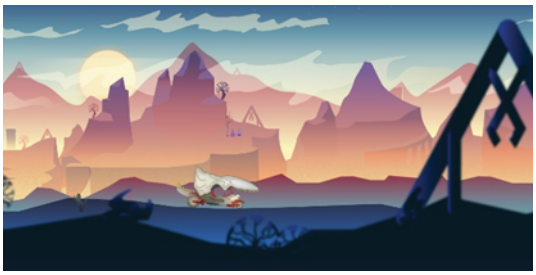

Fig.4. A finished Balance Trucks level.

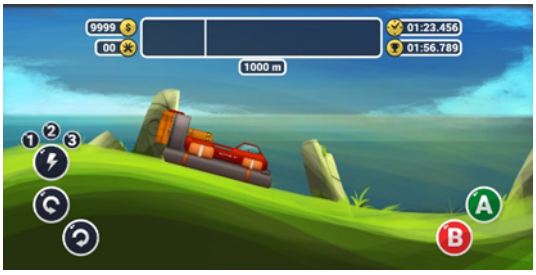

Fig.5. Mock-up of a Balance Trucks level with finalized user-interface.

Looking to the future, the development of the SmartRoadSense gamification still faces a great many challenges. Whilst the gamification, and its associated gameplay, are exceptionally well defined in terms of visual style, a large volume of art assets are still required to ensure that the game environments do not look visually repetitive. Coding of a testbed is in progress, and this will eventually form the basis of the gamification itself. However, as anyone who has worked within the games industry could testify, this will no doubt throw up a whole new range of issues as development progresses, which we will disseminate in follow up publications. Not least of these will be the need to balance the gamification. The degree to which this is performed, and the resulting quality of gameplay, can literally make or break a game, making it key to the success of Balance Trucks.

ACKNOWLEDGMENTS

The SmartRoadSense project would like to thank the following for the creation of artwork: Kimberley Bannister, Emilia Byrne, Nyasha Mhazo, Michael Murphy, Charlotte Palmer, Jia Tan, Vytautas Vasiliuskas, and Lina Vysniauskaite.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687959, and the Disruptive Media Learning Lab, UK.

REFERENCES

[1] J. Togelius, A.J. Champandard, P.L. Lanzi, M. Mateas, A. Paiva, M. Preuss, and K.O. Stanley, "Procedural content generation: Goals, challenges and actionable steps," Artificial and Computational Intelligence in Games, Dagstuhl Follow-Ups, 6, pp. 61-75, 2013.

[2] C. Pedersen, J. Togelius, and G.N. Yannakakis, "Modeling player experience for content creation," IEEE Transactions on Computational Intelligence and AI in Games, 2 (1), pp. 54–67, 2010.

[3] http://www.c4rs.eu/

[4] http://smartroadsense.it/

[5] S. Oliveira, and L. Magalhães, "Adaptive content generation for games", Computação Gráfica e Interação (EPCGI), 2017 24º Encontro Português de, pp. 1-8, 2017.

[6] J. Togelius, E. Kastbjerg, D. Schedl, and G.N. Yannakakis, "What is procedural content generation? Mario on the borderline," in PCGames '11, Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, ACM Press, 2011.

[7] N. Barreto, A. Cardoso, and L. Roque, "Computational creativity in procedural content generation: A state of the art survey," In Proceedings of the 2014 Conference of Science and Art of Video Games, 2014.

[8] J. Togelius, J. Whitehead, and R, Bidarra, "Guest editorial: procedural content generation in games," IEEE Transactions on Computational Intelligence and AI in Games, 3 (3), 2011.

[9] E.J. Hastings, R.K. Guha, and K.O. Stanley, "Automatic content generation in the galactic arms race video game," IEEE Transactions on Computational Intelligence and AI in Games, 1 (4), pp.245-263, 2009.

[10] M. Toy, G. Wichman, K. Arnold, and J. Lane, "Rogue," A.I. Design, 1980.

[11] D. Braben, I. Bell, "Elite," Acornsoft, 1984.

[12] M. Persson, and J. Bergensten, "Minecraft," Mojang, Microsoft Studios, and Sony Computer Entertainment, 2011.

[13] Hello Games, "No Mans Sky," Hello Games, 2016.

[14] [C. Roberts, Star Citizen, Cloud Imperium Games, Unpublished.

[15] J. Togelius, G.N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," IEEE Transactions on Computational Intelligence and AI in Games 3 (3), pp. 172-186, 2011.

[16] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 9 (1), ACM Press, 2013.

[17] R. Khaled, M. J. Nelson, and P. Barr, "Design metaphors for procedural content generation in games," CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems , ACM Press, pp. 1509-1518, 2013.

[18] A.B. Moghadam, and M. K. Rafsanjani, "A genetic approach in procedural content generation for platformer games level creation," 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), 2017.

[19] M. Kerssemakers, J. Tuxen, J. Togelius, G.N. Yannakakis, "A procedural procedural level generator generator," 2012 IEEE Conference on Computational Intelligence and Games (CIG), 2012.

[20] J. Dormans, "Adventures in level design: Generating missions and spaces for action adventure games," PCGames '10 Proceedings of the 2010 Workshop on Procedural Content Generation in Games, ACM Press, 2010.

[21] M. Nitsche, C. Ashmore, W, Hankinson, R. Fitzpatrick, J. Kelly, and K. Margenau, "Designing procedural game spaces: A case study," in Proceedings of FuturePlay, 2006

[22] J. Togelius, and J. Schmidhuber, “An experiment in automatic game design,” IEEE Symposium On Computational Intelligence and Games, CIG '08, 2008.

[23] J. Togelius, S. Karakovskiy, J. Koutník, and J. Schmidhuber, “Super Mario Evolution,” 2009 IEEE Symposium on Computational Intelligence and Games, CIG2009, pp.156-161, 2009.

[24] J. Togelius, S. Karakovskiy, and R. Baumgarten, “The 2009 Mario AI Competition,” IEEE Congress on Evolutionary Computation (CEC), pp. 1-8, 2010.

[25] https://ec.europa.eu/programmes/horizon2020/

[26] http://www.openstreetmap.org/

[27] V. Freschi, S. Delpriori, L. Klopfenstein, E. Lattanzi, G. Luchetti, and A. Bogliolo, “Geospatial data aggregation and reduction in vehicular sensing applications: The case of road surface monitoring,” International Conference on Connected Vehicles and Expo. ICCVE, 2014.

[28] G. Alessandroni, L. Klopfenstein, S. Delpriori, M. Dromedari, G. Luchetti, B. Paolini, A. Seraghiti, E. Lattanzi, V. Freschi, A. Carini, and A. Bogliolo, “Smartroadsense: Collaborative road surface condition monitoring,” in Proceedings of the Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Ubicomm 2014. IARIA, 2014.

[29] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, “The pothole patrol: Using a mobile sensor network for road surface monitoring,” in Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services. ACM Press, pp. 29–39, 2008.

[30] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich monitoring of road and traffic conditions using mobile smartphones,” in Proc. of the 6th ACM Conference on Embedded Network Sensor Systems. ACM Press, pp. 323–336, 2008.

[31] A. Thiagarajan et al., “VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones,” in Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems. ACM Press, pp. 85–98, 2009.

[32] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, “Accurate, low-energy trajectory mapping for mobile devices,” in Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’11. USENIX Association, pp. 267–280, 2011.

[33] K. Aberer et al., “Opensense: open community driven sensing of environment,” in Proceedings of the 1st International Workshop on GeoStreaming (IWGS ’10), pp. 39–42, 2010.

[34] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, “Wolverine: Traffic and road condition estimation using smartphone sensors,” in Proceedings of COMSNETS ’12, pp. 1–6, 2012.

[35] K. Chen, M. Lu, G. Tan, and J. Wu, “CRSM: Crowdsourcing based road surface monitoring,” in Proceedings of HPCC EUC ’13, pp. 2151–2158, 2013.

[36] V. Douangphachanh and H. Oneyama, “A study on the use of smartphones for road roughness condition estimation,” Journal of the Eastern Asia Society for Transportation Studies, vol. 10, no. 0, pp. 1551–1564, 2013.

[37] S. Clarke, S. Arnab, M. Lewis, L. Morini, S. Depriori, A. Bogliolo, & L. Klopfenstein, “A gamified approach for facilitating a user-engagement strategy for public-led collective awareness platform for road sensing,” in Proceedings of the 11th European Conference on Games Based Learning, ECGBL 2017 (pp. 79-87). Academic Conferences and Publishing International Limited, 2017.

[38] G. Smith, “Understanding procedural content generation: A design-centric analysis of the role of PCG in games”, Proceedings of the ACM CHI Conference on Human Factors in Computing Systems (CHI 2014), 2014.

[39] A. Sullivan, “Content Selection vs. Content Generation”, Expressive Intelligence Studio, https://eis-blog.soe.ucsc.edu/2010/06/content-selection-vs-content-generation/, 2010.

[40] S. Miyamoto, Super Mario Kart, Nintendo Co. Ltd, 1992.